DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Assignment - 4

Student Name: Subhojit Ghosh **UID: 22BCS15368**

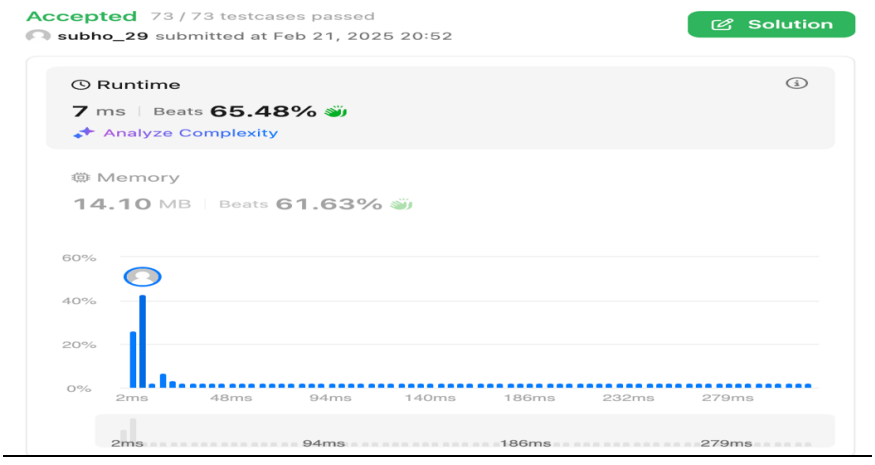
Branch: CSE **Section/Group:FL-602-A**

Semester: 6 **Date of Performance:21.02.2025**

Subject Name: Advanced Programming **Subject Code: 22CSH-359**

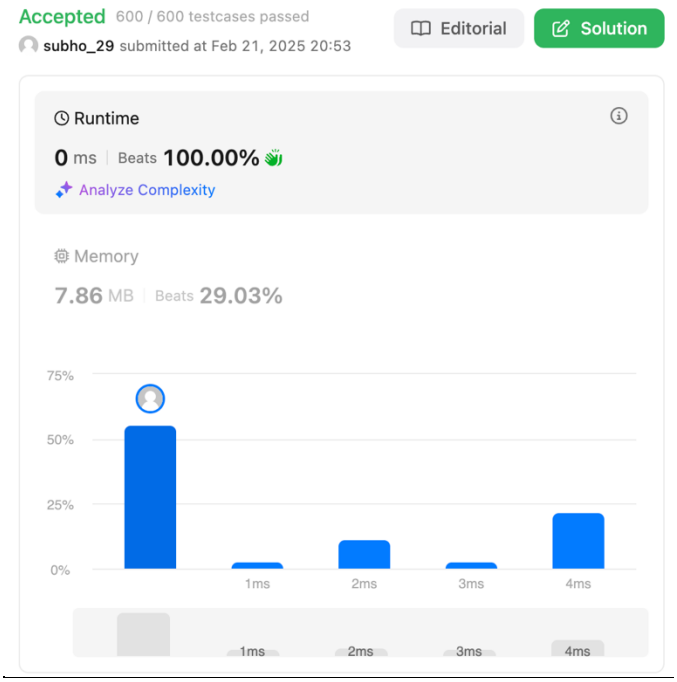
1. **Longest Nice Substring**

```
class Solution {
public:
    string longestNiceSubstring(string s) {
         if (s.size() < 2) return "";
        unordered_set<char> st(begin(s), end(s));
        for (int i = 0; i < s.size(); i++) {
             if (st.find((char) toupper(s[i])) == end(st) ||
st.find((char)tolower(s[i])) == end(st)) {
                string s1 = longestNiceSubstring(s.substr(0, i));
                string s2 = longestNiceSubstring(s.substr(i + 1));
                return s1.size() >= s2.size() ? s1 : s2;
            }
        }
        return s;
    }
};
```



2. Reverse Bits

```cpp
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        n = ((n & 0xffff0000) >> 16) | ((n & 0x0000ffff) << 16);
        n = ((n & 0xff00ff00) >> 8) | ((n & 0x00ff00ff) << 8);
        n = ((n & 0xf0f0f0f0) >> 4) | ((n & 0x0f0f0f0f) << 4);
        n = ((n & 0xcccccccc) >> 2) | ((n & 0x33333333) << 2);
        n = ((n & 0xaaaaaaaa) >> 1) | ((n & 0x55555555) << 1);

        return n;

    }
};
```



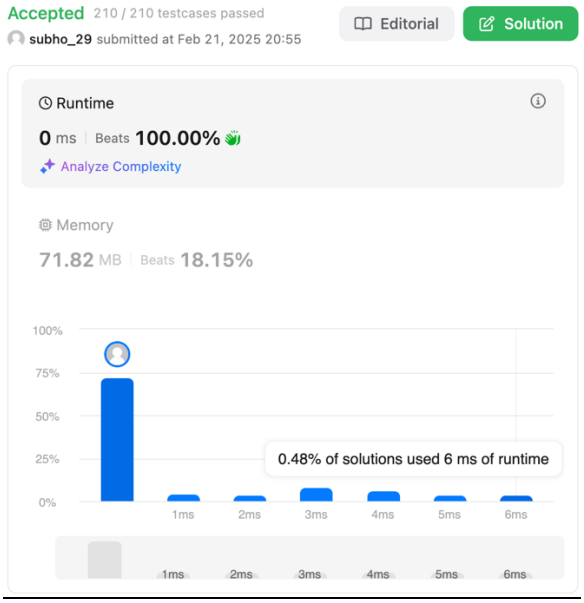
3. Number of 1 Bits

```cpp
class Solution {
public:
    int hammingWeight(int n) {
         int res = 0;
        for (int i = 0; i < 32; i++) {
            if ((n >> i) & 1) {
                res += 1;
            }
        }
        return res;
    }
```

};

4. Maximum Subarray

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
         int maxSum = INT_MIN;
        int currentSum = 0;

        for (int i = 0; i < nums.size(); i++) {
            currentSum += nums[i];

            if (currentSum > maxSum) {
                maxSum = currentSum;
            }

            if (currentSum < 0) {
                currentSum = 0;
            }
        }

        return maxSum;

    }
};
```



5. [Search a 2D Matrix II]

```cpp
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int n = matrix.size(), m = matrix[0].size();
        int row = 0, col = m - 1;

        while (row < n && col >= 0) {
            if (matrix[row][col] == target) return true;
            else if (matrix[row][col] < target) row++;
            else col--;
        }
        return false;
    }
};
```

6. Super Pow

```cpp
class Solution {
    const int base = 1337;
    int powmod(int a, int k) //a^k mod 1337 where 0 <= k <= 10
    {
        a %= base;
        int result = 1;
        for (int i = 0; i < k; ++i)
            result = (result * a) % base;
        return result;
    }
public:
    int superPow(int a, vector<int>& b) {
        if (b.empty()) return 1;
        int last_digit = b.back();
        b.pop_back();
        return powmod(superPow(a, b), 10) * powmod(a, last_digit) % base;
    }
};
```